D212 Performance Assessment - Association Rules and Lift Analysis

Name: Coots, Anthony.

Affiliation: Grad Student M.Sc Data Analytics.

Date: `2024-06-05`

Version: 1.1.5, r1.

Introduction

"In this task, you will act as an analyst and create a data mining report. You must select one of the data dictionary and data set files to use for your report from the following web link: "Data Sets and Associated Data Dictionaries.""

- WGU

Competencies

4030.6.6 : Pattern Predictions

- The graduate predicts patterns in data using association rules and lift analysis.

Scenario

"One of the most critical factors in patient relationship management that directly affects a hospital's long-term cost-effectiveness is understanding the patients and the conditions leading to hospital admissions. When a hospital understands its patients' characteristics, it is better able to target treatment to patients, resulting in a more effective cost of care for the hospital in the long term.

You are an analyst for a hospital that wants to better understand the characteristics of its patients. You have been asked to perform a market basket analysis to analyze patient data to identify key associations of your patients, ultimately enabling better business and strategic decision-making for the hospital."

- WGU

Table of Contents:

Research Question

A1: Proposal of Question

Propose one question that can be answered using market basket analysis and is relevant to a real-world organizational situation.

Market Basket analysis can be used to determine patterns of prescriptions that are statistically backed by metric measurements. Metrics such as support, confidence and lift help provide insight based on antecedent and consequent, or if-then structure. Support, to help determine how frequently a combination of prescriptions occur. Confidence, to provide insight into the likeliness of drug or vitamin B being prescribed if drug or vitamin A was. Lastly lift, to observe the frequency of drug, vitamin, etc A and B being prescribed together with the frequency as if they were independent of one another. By using these metrics to facilitate Market Basket analysis, the following question is proposed:

Question: "*What prescriptions are associated with patients based on support, confidence and lift in order to effectively determine prescription patterns for hospital insights*?"

A2: Defined Goal

Define one reasonable goal of the data analysis that is within the scope of the scenario and is represented in the available data.

The goal of the Market Basket analysis is to determine prescription patterns of the twenty available prescription columns (Presc01-Presc20). The patterns found among these prescriptions could provide important insight for hospital operations such as common treatment patterns, inventory management and more. Using metrics such as support, confidence and lift, then later multimetric filtering will be used to identify the patterns between all antecedents and consequents.

Market Basket Justification

B1: Explanation of Market Basket

Logically explain how market basket analyzes the selected data set and include expected outcomes.

Market Basket analysis at it's foundation, identifies patterns based on historical data. With the twenty prescription variables available, this analysis finds these patterns based on 'if-then' or antecedent consequent structure. For example, if medication A is prescribed, then medication B is also prescribed. The frequency of this combination is the make up of the 'support' metric. 'Confidence,' then measures the likeliness of medication B being prescribed when medication A is. Lastly, 'lift' addresses the frequency of medications A and B being together only relative to the independent frequencies of medications A and B alone. These metrics are used to facilitate Market Basket analysis. The expected outcomes of this analysis identify the prescription patters being filtered by multimetric considerations (minimum threshold scores, minimum support, single antecedents, etc) which may provide insight into patient care and hospital operation.

B2: Transaction Example

Include one accurate example of transactions in the data set.

58	abilify	~~nphetamine salt combo xr~~	~~clopidogrel~~	diazepam	glyburide
59					
60	metoprolol	carvedilol	mometasone	abilify	
61					
62	methylprednisone	potassium Chloride	salmeterol inhaler	celebrex	
63					
64	abilify	diazepam	allopurinol	nphetamine salt combo xr	

The above image includes transactions from the data set used in the analysis. For example, the prescriptions 'Abilify' and 'Diazepam' appear together in multiple transactions. This pairing could suggest a potential pattern where both medications are prescribed commonly which may indicate a potential common treatment pattern or could be used for inventory arrangement/management for easier patient assessment. However, to confirm the significance of this pattern it is important to seek the support, confidence, and lift metrics to determine such significance of this "pattern."

B3: Market Basket Assumption

Summarize one assumption of market basket analysis.

Market Basket analysis can use a couple of different algorithms under the hood, but for this analysis the '*Apriori*' algorithm is used. The '*Apriori*' algorithm assumes it's key principle otherwise known as the '*Apriori principle*.' This principle, or assumption, is that if an itemset is infrequent then all of its supersets are also infrequent. The converse is also assumed. This assumption is important because by limiting the number of itemsets the algorithm needs to examine will reduce the computational complexity thus allowing the '*Apriori*' algorithm to generate the association rules more efficiently. The association rules that can be used to answer the question in A1, using metrics like support, confidence and lift.

Data Preparation and Analysis

C1: Transforming the Data Set

Transform the data set to make it suitable for market basket analysis.

In [1]:
```
# Needed install.
```

In [2]:
```
pip install mlxtend
```

```
Requirement already satisfied: mlxtend in c:\users\acoots\appdata\local\anaconda3\l
ib\site-packages (0.23.1)
Requirement already satisfied: scipy>=1.2.1 in c:\users\acoots\appdata\local\anacon
da3\lib\site-packages (from mlxtend) (1.11.4)
Requirement already satisfied: numpy>=1.16.2 in c:\users\acoots\appdata\local\anaco
nda3\lib\site-packages (from mlxtend) (1.26.4)
Requirement already satisfied: pandas>=0.24.2 in c:\users\acoots\appdata\local\anac
onda3\lib\site-packages (from mlxtend) (2.1.4)
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\acoots\appdata\local
\anaconda3\lib\site-packages (from mlxtend) (1.2.2)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\acoots\appdata\local\a
naconda3\lib\site-packages (from mlxtend) (3.8.0)
Requirement already satisfied: joblib>=0.13.2 in c:\users\acoots\appdata\local\anac
onda3\lib\site-packages (from mlxtend) (1.2.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\acoots\appdata\local\an
aconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\acoots\appdata\local\anacon
da3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\acoots\appdata\local\a
naconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\acoots\appdata\local\a
naconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\acoots\appdata\local\ana
conda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\acoots\appdata\local\anaco
nda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\acoots\appdata\local\an
aconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\acoots\appdata\loca
l\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\acoots\appdata\local\anacon
da3\lib\site-packages (from pandas>=0.24.2->mlxtend) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\acoots\appdata\local\anac
onda3\lib\site-packages (from pandas>=0.24.2->mlxtend) (2023.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\acoots\appdata\loca
l\anaconda3\lib\site-packages (from scikit-learn>=1.0.2->mlxtend) (2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\acoots\appdata\local\anaconda3
\lib\site-packages (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
# Import list.
import matplotlib.pyplot as plt
from mlxtend.frequent_patterns import apriori, association_rules
from mlxtend.preprocessing import TransactionEncoder
import numpy as np
import os
import pandas as pd
import seaborn as sns
```

In [4]:
```
# What is my current working directory?
print("\n\n Current Working Directory: " + os.getcwd() + '\n')
```

```


 Current Working Directory: C:\Users\acoots\Desktop\Personal\Education\WGU\Data Ana
lytics, M.S\D212 - Data Mining II\Task 3 - Association Rules and Lift Analysis

```

In [5]:
```
# Read data into DataFrame.
df = pd.read_csv("medical_market_basket.csv")
```

In [6]:
```
# Display current state.
print(df.head())
```

```
      Presc01            Presc02                   Presc03       Presc04  \
0         NaN                NaN                       NaN           NaN
1  amlodipine  albuterol aerosol               allopurinol  pantoprazole
2         NaN                NaN                       NaN           NaN
3  citalopram            benicar  amphetamine salt combo xr           NaN
4         NaN                NaN                       NaN           NaN

     Presc05     Presc06     Presc07      Presc08     Presc09      Presc10  \
0        NaN         NaN         NaN          NaN         NaN          NaN
1  lorazepam  omeprazole  mometasone  fluconozole  gabapentin  pravastatin
2        NaN         NaN         NaN          NaN         NaN          NaN
3        NaN         NaN         NaN          NaN         NaN          NaN
4        NaN         NaN         NaN          NaN         NaN          NaN

  Presc11   Presc12                 Presc13           Presc14  Presc15  \
0     NaN       NaN                     NaN               NaN      NaN
1  cialis  losartan  metoprolol succinate XL  sulfamethoxazole  abilify
2     NaN       NaN                     NaN               NaN      NaN
3     NaN       NaN                     NaN               NaN      NaN
4     NaN       NaN                     NaN               NaN      NaN

          Presc16        Presc17       Presc18       Presc19    Presc20
0             NaN            NaN           NaN           NaN        NaN
1  spironolactone  albuterol HFA  levofloxacin  promethazine  glipizide
2             NaN            NaN           NaN           NaN        NaN
3             NaN            NaN           NaN           NaN        NaN
4             NaN            NaN           NaN           NaN        NaN
```

In [7]:
```
# Every other row is completely empty in the data set, we'll drop those rows.
df = df.dropna(how = 'all')
# Reset row number.
df = df.reset_index(drop = True)
```

In [8]:
```
print(df.head())
```

```
      Presc01            Presc02                     Presc03       Presc04  \
0  amlodipine  albuterol aerosol                 allopurinol  pantoprazole
1  citalopram            benicar  amphetamine salt combo xr           NaN
2   enalapril                NaN                         NaN           NaN
3  paroxetine        allopurinol                         NaN           NaN
4     abilify       atorvastatin                  folic acid      naproxen

     Presc05     Presc06     Presc07     Presc08     Presc09      Presc10  \
0  lorazepam  omeprazole  mometasone  fluconozole  gabapentin  pravastatin
1        NaN         NaN         NaN          NaN         NaN          NaN
2        NaN         NaN         NaN          NaN         NaN          NaN
3        NaN         NaN         NaN          NaN         NaN          NaN
4   losartan         NaN         NaN          NaN         NaN          NaN

  Presc11   Presc12                 Presc13           Presc14  Presc15  \
0  cialis  losartan  metoprolol succinate XL  sulfamethoxazole  abilify
1     NaN       NaN                      NaN               NaN      NaN
2     NaN       NaN                      NaN               NaN      NaN
3     NaN       NaN                      NaN               NaN      NaN
4     NaN       NaN                      NaN               NaN      NaN

          Presc16        Presc17       Presc18       Presc19    Presc20
0  spironolactone  albuterol HFA  levofloxacin  promethazine  glipizide
1             NaN            NaN           NaN           NaN        NaN
2             NaN            NaN           NaN           NaN        NaN
3             NaN            NaN           NaN           NaN        NaN
4             NaN            NaN           NaN           NaN        NaN
```

In [16]:
```
# Single list of prescriptions per transaction.
transactions = df.apply(lambda row: [item for item in row.dropna().unique()], axis
print(transactions.head(5))

# TransactionEncoder initialization.
encoder = TransactionEncoder()

# Fit and transform the data to an array of boolean values.
onehot = encoder.fit(transactions).transform(transactions)

# Convert the boolean array to a DataFrame
basket = pd.DataFrame(onehot, columns=encoder.columns_)
```

```
0    [amlodipine, albuterol aerosol, allopurinol, p...
1     [citalopram, benicar, amphetamine salt combo xr]
2                                          [enalapril]
3                             [paroxetine, allopurinol]
4    [abilify, atorvastatin, folic acid, naproxen, ...
dtype: object
```

In [17]:
```
# Show results of encoded data.
print(basket)
```

```
      Duloxetine  Premarin    Yaz  abilify  acetaminophen  actonel  \
0          False     False  False     True          False    False
1          False     False  False    False          False    False
2          False     False  False    False          False    False
3          False     False  False    False          False    False
4          False     False  False     True          False    False
...          ...       ...    ...      ...            ...      ...
7496       False     False  False    False          False    False
7497       False     False  False    False          False    False
7498       False     False  False    False          False    False
7499       False     False  False    False          False    False
7500       False     False  False    False          False    False

      albuterol HFA  albuterol aerosol  alendronate  allopurinol  ...  \
0              True               True        False         True  ...
1             False              False        False        False  ...
2             False              False        False        False  ...
3             False              False        False         True  ...
4             False              False        False        False  ...
...             ...                ...          ...          ...  ...
7496          False              False        False        False  ...
7497          False              False        False        False  ...
7498          False              False        False        False  ...
7499          False              False        False        False  ...
7500          False              False        False        False  ...

      trazodone HCI  triamcinolone Ace topical  triamterene  trimethoprim DS  \
0             False                      False        False            False
1             False                      False        False            False
2             False                      False        False            False
3             False                      False        False            False
4             False                      False        False            False
...             ...                        ...          ...              ...
7496          False                      False        False            False
7497          False                      False        False            False
7498          False                      False        False            False
7499          False                      False        False            False
7500          False                      False        False            False

      valaciclovir  valsartan  venlafaxine XR  verapamil SR  viagra  zolpidem
0            False      False           False         False   False     False
1            False      False           False         False   False     False
2            False      False           False         False   False     False
3            False      False           False         False   False     False
4            False      False           False         False   False     False
...            ...        ...             ...           ...     ...       ...
7496         False      False           False         False   False     False
7497         False      False           False         False   False     False
7498         False      False           False         False   False     False
7499         False      False           False         False   False     False
7500         False      False           False         False   False     False

[7501 rows x 119 columns]
```

In [12]:
```
# Export cleaned dataset to csv.
basket.to_csv("analysis_ready_medical_clean.csv", index = False)
```

C2: Code Execution

Execute the code used to generate association rules with the Apriori algorithm.

In [13]:
```
# Prevalence of an itemset in all transactions, large data set therefore 0.02 (2%)
min_support = 0.02
# 1% found ~ 440 patterns, likely too much.

# Apriori algorithm determining the frequent itemsets.
frequent_itemsets = apriori(basket, min_support = min_support, use_colnames = True

# Data set for association rules based on frequent itemsets determined by apriori.
rules = association_rules(
    frequent_itemsets,
    metric = "lift",
    min_threshold = 0.01
)

# General values for rules.
print(rules)
```

```
                  antecedents              consequents  antecedent support  \
0                 (amlodipine)                (abilify)            0.071457   
1                    (abilify)             (amlodipine)            0.238368   
2       (amphetamine salt combo)              (abilify)            0.068391   
3                    (abilify)  (amphetamine salt combo)            0.238368   
4    (amphetamine salt combo xr)              (abilify)            0.179709   
..                         ...                      ...                 ...   
95                (metoprolol)               (diazepam)            0.095321   
96         (doxycycline hyclate)            (glyburide)            0.095054   
97                 (glyburide)    (doxycycline hyclate)            0.170911   
98                 (glyburide)              (losartan)            0.170911   
99                  (losartan)             (glyburide)            0.132116   

    consequent support   support  confidence      lift  leverage  conviction  \
0             0.238368  0.023597    0.330224  1.385352  0.006564    1.137144   
1             0.071457  0.023597    0.098993  1.385352  0.006564    1.030562   
2             0.238368  0.024397    0.356725  1.496530  0.008095    1.183991   
3             0.068391  0.024397    0.102349  1.496530  0.008095    1.037830   
4             0.238368  0.050927    0.283383  1.188845  0.008090    1.062815   
..                 ...       ...         ...       ...       ...         ...   
95            0.163845  0.022930    0.240559  1.468215  0.007312    1.101015   
96            0.170911  0.020131    0.211781  1.239135  0.003885    1.051852   
97            0.095054  0.020131    0.117785  1.239135  0.003885    1.025766   
98            0.132116  0.028530    0.166927  1.263488  0.005950    1.041786   
99            0.170911  0.028530    0.215943  1.263488  0.005950    1.057436   

    zhangs_metric  
0        0.299568  
1        0.365218  
2        0.356144  
3        0.435627  
4        0.193648  
..            ...  
95       0.352502  
96       0.213256  
97       0.232768  
98       0.251529  
99       0.240286  

[100 rows x 10 columns]
```

C3: Association Rules Table

Include values for the support, lift and confidence of the association rules table.

In [14]:
```
# Association Rules Table.
# Sorted results by confidence. High confidence, consquent is likely present when
association_rules_table = rules.sort_values(by = "confidence", ascending = False)
print(association_rules_table)
```

```
      antecedents     consequents  antecedent support  consequent support  \
35    (metformin)       (abilify)            0.050527            0.238368
25    (glipizide)       (abilify)            0.065858            0.238368
31   (lisinopril)       (abilify)            0.098254            0.238368
81   (lisinopril)    (carvedilol)            0.098254            0.174110
22  (fenofibrate)       (abilify)            0.051060            0.238368
..            ...             ...                 ...                 ...
34      (abilify)     (metformin)            0.238368            0.050527
14      (abilify)   (clopidogrel)            0.238368            0.059992
29      (abilify)  (levofloxacin)            0.238368            0.063325
23      (abilify)   (fenofibrate)            0.238368            0.051060
39      (abilify)      (naproxen)            0.238368            0.058526

     support  confidence      lift  leverage  conviction  zhangs_metric
35  0.023064    0.456464  1.914955  0.011020    1.401255       0.503221
25  0.027596    0.419028  1.757904  0.011898    1.310962       0.461536
31  0.040928    0.416554  1.747522  0.017507    1.305401       0.474369
81  0.039195    0.398915  2.291162  0.022088    1.373997       0.624943
22  0.020131    0.394256  1.653978  0.007960    1.257349       0.416672
..       ...         ...       ...       ...         ...            ...
34  0.023064    0.096756  1.914955  0.011020    1.051182       0.627330
14  0.022797    0.095638  1.594172  0.008497    1.039415       0.489364
29  0.020264    0.085011  1.342461  0.005169    1.023701       0.334938
23  0.020131    0.084452  1.653978  0.007960    1.036472       0.519145
39  0.020131    0.084452  1.442993  0.006180    1.028318       0.403076

[100 rows x 10 columns]
```

C4: Top Three Rules

Include the top 3 relevant rules and explain them.

In [15]:
```
print(association_rules_table.head(3))
```

```
     antecedents consequents  antecedent support  consequent support  \
35   (metformin)   (abilify)            0.050527            0.238368   
25   (glipizide)   (abilify)            0.065858            0.238368   
31  (lisinopril)   (abilify)            0.098254            0.238368   

     support  confidence      lift  leverage  conviction  zhangs_metric  
35  0.023064    0.456464  1.914955  0.011020    1.401255       0.503221  
25  0.027596    0.419028  1.757904  0.011898    1.310962       0.461536  
31  0.040928    0.416554  1.747522  0.017507    1.305401       0.474369  
```

Rule 1:

This rule suggests that there exists a strong relationship between the antecedent '*metformin*' and the consequent '*abilify*.' The support claims that roughly 2% of all transactions in the data set contain both '*metformin*' and '*abilify*.' A confidence of about 46% of all transactions including '*metformin*' includes an '*abilify*' prescription. Lastly, the lift suggests that '*abilify*' is nearly 2 times as likely to be prescribed with '*metformin*' than the two prescriptions being independent.

Rule 2:

Similar to the prior, this rule suggests that there exists a strong relationship between the antecedent '*glipizide*' and the consequent '*abilify*.' The support claims that roughly 3% of all transactions in the data set contain both '*glipizide*' and '*abilify*.' A confidence of about 42% of all transactions including '*glipizide*' includes an '*abilify*' prescription. Lastly, the lift suggests that '*abilify*' is 1.75 times as likely to be prescribed with '*glipizide*' than the two prescriptions being independent.

Rule 3:

Like the previous 2, this rule suggests that there exists a strong relationship between medication A, the antecedent '*lisinopril*' and medication B, the consequent '*abilify*.' The support claims that roughly 4% of all transactions in the data set contain both '*lisinopril*' and '*abilify*.' A confidence of almost 42% of all transactions including '*lisinopril*' includes an '*abilify*' prescription. Lastly, the lift suggests that '*abilify*' is 1.74 times as likely to be prescribed with '*lisinopril*' than the two prescriptions being independent.

Data Summary and Implications

D1: Significance of Support, Lift, and Confidence Summary

Summarize the significance of support, lift, and confidence from the results.

Support:

Support indicates how frequently the itemset appears within the data set. In the prior rules, support shows the percentage of transactions that include the antecedents (e.g. '*metformin*,' '*glipizide*' and '*lisinopril*') and coincidentally their similar consequent ('*abilify*.') Of the ~7,500 records available in the cleaned data set, the combination of the antecedent and consequent make up 2%, 3% and 4% of the data set combinations in totality, respectively.

Lift:

Lift determines the strength of the associations, meaning the ratio of the support of the combination against the support for each prescription independently. For example, a lift greater than 1.0 as seen in the top three rules suggest that the probability of the presciptions together are more likely than independently. For example, '*metformin*' and '*abilify*' had a lift of 1.91, higher than a score of 1, indicating the prior. If certain antecendent and consequents are more likely together, this may be important for integrated strategy for the hospital.

Confidence:

Confidence measures the likeliness that the consequent, in this case '*abilify*', is found in transactions containing the antecedent. For example, a confidence of 45.47% between '*metformin*' and '*abilify*' suggests that almost half of the patients on the antecedent are taking the consequent. The higher the confidence, the stronger a hospital should consider the decisions or details behind why this correlation exists.

D2: Practical Significance of Findings

Discuss the practical significance of the findings.

For this analysis, the findings can imply which medications are commonly prescribed together and thus build foundation for conditions that require both the antecedent and consequent in health management. Additionally, hospitals and clinics can potentially manage their inventory better knowing that these medications frequently go together such as having '*metformin*,' '*glipizide*,' '*lisinopril*' and '*abilify*' near one another. Finally, insights from understanding which prescriptions go together can derive individual policies. For example, if a patient requires '*metformin*' then, tests are likely needed to determine if they require any of the prior top three antecedents.

D3: Course of Action

Recommend a course of action for the real-world organizational situation.

Considering the results of the analysis, the following are recommended actions for the hospital chain. Firstly, healthcare providers ideally develop studies into the effects and safety of prescription combinations such as '*metformin*,' '*glipizide*,' '*lisinopril*' and '*abilify*' or the combinations seen from the data analysis. The context of patient prescription is the critical as there is no one-fits-all solution medically for all patients. After facilitating these studies, should policies or even suggestions come to fruition then all appropriate medical staff should be trained or informed of what is to come.

Additionally, inventory management should be implemented in order to have appropriate supply if needed at the hospital, some prescriptions are external to the hospital while some come directly from the hospital itself. When a pharmacy is running low on the needed drug combinations then actions should be taken to prevent being out of the needed medications. This analysis should continuously applied to new data, at least within 2 years, in order to always keep tabs on what combinations of medications develop or even dissipate.

Attachments

E: Panopto Recording

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=35f71ba8-7b6d-4329-bf49-b1860109ba2e

F: Sources for Third-Party Code

DataCamp Course Resource.

G: Sources

DataCamp Course Resouce.

Overload, D. (Ed.). (2023, March 9). Market Basket Analysis: Techniques, Applications, and Benefits for Retailers. Medium. https://medium.com/@data-overload/market-basket-analysis-techniques-applications-and-benefits-for-retailers-d66eed1f917e